CLAIMS

What is claimed is:

1. A method for simultaneous debugging of an electrical design having both an HDL portion and a general programming language portion, comprising:

interrupting a simulator handling the HDL portion, the simulator interrupted by an external debugger, the external debugger for debugging the general language portion;

handling a simulator request with an external debugger, the external debugger calling a request processing function at the simulator; and

executing the request processing function at the simulator to respond to the simulator request.

2. The method of claim 1 in which the simulator request accesses a portion of the HDL portion.

3. The method of claim 2 in which the simulator request accesses HDL signal values.

4. The method of claim 2 in which the simulator request accesses HDL design hierarchy.

5. The method of claim 1 in which the simulator request operates simulator functionality.

6. The method of claim 1 in which the general programming language portion comprises C, C++, or SystemC code.

7. The method of claim 1 in which the HDL portion comprises VHDL or Verilog.

8. The method of claim 1 in which the action of having the external debugger call the request processing function is based upon recognition of a waiting simulator request.

9. The method of claim 8 in which recognition of the waiting simulator request is based upon a message sent to the external debugger.

10. The method of claim 8 in which recognition of the waiting simulator request is based upon a periodic check of a simulator request wait queue.

11. The method of claim 8 in which recognition of the waiting simulator request is based on whether a threshold number of simulator requests are waiting in a simulator request wait queue.

12. The method of claim 1 in which the simulator request is generated at a simulator GUI.

13. The method of claim 12 in which the response to the simulator request is displayed at the simulator GUI.

14. The method of claim 1 in which the external debugger calls the request processing function at the simulator with the following statement:

call *expr*

15. The method of claim 1 in which the simulator request is routed through a debugger GUI for the external debugger.

16. The method of claim 1 in which the simulator request is directly routed to the external debugger.

17. The method of claim 1 in which the request processing function is set up ahead of time at the simulator to handle anticipated simulator requests.

18. A method for simultaneous processing of a design that is based upon multiple programming languages, the multiple programming languages comprising a first language portion and a second language portion, in which processing of the second language portion interrupts processing of the first language portion, the method comprising:

processing the second language portion of the design causing an interruption of processing for the first language portion;

determining whether there are one or more waiting requests for processing of the first language portion;

handling the one or more waiting requests for processing of the first language portion by having processing of the second language portion call a request processing function at the first language portion; and

executing the request processing function at the first language portion to process the one or more waiting requests.

19. The method of claim 18 in which the one or more waiting requests are for accessing data from the first language portion of the design.

20. The method of claim 18 in which the one or more waiting requests are for debugging the first language portion.

21. The method of claim 18 the act of determining whether there are one or more waiting requests for processing of the first language portion is based upon a message sent to the processing of the second language portion.

22. The method of claim 18 the act of determining whether there are one or more waiting requests for processing of the first language portion is based a periodic check of a request wait queue for the first language portion.

23. The method of claim 18 the act of determining whether there are one or more waiting requests for processing of the first language portion is based on whether a threshold number of simulator requests are waiting in a request wait queue.

24. The method of claim 18 in which the request processing function is called with the following statement:

call *expr*

25. The method of claim 18 in which processing the second language portion comprises debugging the second language portion.

26. The method of claim 18 in which the request processing function is set up ahead of time to handle anticipated requests.

27. A computer program product comprising a computer usable medium having executable code to execute a process for simultaneous debugging of an electrical design having both an HDL portion and a general programming language portion, the process comprising:

interrupting a simulator handling the HDL portion, the simulator interrupted by an external debugger, the external debugger for debugging the general language portion;

handling a simulator request with an external debugger, the external debugger calling a request processing function at the simulator; and

executing the request processing function at the simulator to respond to the simulator request.

28. A system for simultaneous debugging of an electrical design having both an HDL portion and a general programming language portion, comprising:

means for interrupting a simulator handling the HDL portion, the simulator interrupted by an external debugger, the external debugger for debugging the general language portion;

means for handling a simulator request with an external debugger, the external debugger calling a request processing function at the simulator; and

means for executing the request processing function at the simulator to respond to the simulator request.

29. A computer program product comprising a computer usable medium having executable code to execute a method for simultaneous processing of a design that is based upon multiple programming languages, the multiple programming languages comprising a first language portion and a second language portion, in which processing of the second language portion interrupts processing of the first language portion, the method comprising:

processing the second language portion of the design causing an interruption of processing for the first language portion;

determining whether there are one or more waiting requests for processing of the first language portion;

handling the one or more waiting requests for processing of the first language portion by having processing of the second language portion call a request processing function at the first language portion; and

executing the request processing function at the first language portion to process the one or more waiting requests.

30. A system for simultaneous processing of a design that is based upon multiple programming languages, the multiple programming languages comprising a first language portion and a second language portion, in which processing of the second language portion interrupts processing of the first language portion, the method comprising:

means for processing the second language portion of the design causing an interruption of processing for the first language portion;

means for determining whether there are one or more waiting requests for processing of the first language portion;

means for handling the one or more waiting requests for processing of the first language portion by having processing of the second language portion call a request processing function at the first language portion; and

means for executing the request processing function at the first language portion to process the one or more waiting requests.